

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR AUTOMATICALLY DETERMINING  
A PEAK VOLTAGE LEVEL FOR A DATA SIGNAL PROPAGATED ON A  
CARRIER MEDIUM

Inventors:

Jeremiah M. Jeffress

Prepared by:

Blakely, Sokoloff, Taylor & Zafman LLP  
12400 Wilshire Boulevard, Seventh Floor  
Los Angeles, California 90025-1026  
(408) 720-8598

Docket No. 085710.P036

"Express Mail" mailing label number: EM564109361US

Date of Deposit: September 15, 1998

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service  
"Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee  
has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Lindy Vajretti

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

(Date signed)

085710.P036

**METHOD AND APPARATUS FOR AUTOMATICALLY DETERMINING  
A PEAK VOLTAGE LEVEL FOR A DATA SIGNAL PROPAGATED ON A  
CARRIER MEDIUM**

5    **FIELD OF THE INVENTION**

        The present invention relates generally to the field of data networking. More specifically, the present invention relates to a method and apparatus for automatically determinng a peak voltage level for a data signal propagated on a carrier medium.

10

**BACKGROUND OF THE INVENTION**

        Nodes comprising a data network may transmit and receive information over a guided media, such as metal wire or fiber-optic cable, or an unguided media, such as by infrared or microwave. Copper cable is  
15    a relatively inexpensive, well-understood technology that is easily installed and is accordingly the cable of choice in a majority of network installations. However, copper cable suffers from various electrical characteristics that impose transmission limits. Types of copper cable that may be used to transmit digital signals comprise straight cable (i.e.,  
20    copper wire surrounded by an insulated sheath), twisted-pair cable (i.e., a pair of copper wires that are twisted together to former pair) and coaxial cable (i.e., a solid copper core surrounded by insulator, a combination shield and ground wire, and an outer protective jacket).

A recent and emerging method of implementing a network within a home, office or multi-unit dwelling proposes utilizing existing and installed telephone cable within the structure as the guided medium via which nodes may transmit and receive information. Such telephone  
5 cable (a.k.a. Plain Old Telephone Service (POTS) wiring) is often Category 1 or Category 2 Unshielded Twisted-Pair (UTP) telephone cable, and may accordingly be particularly susceptible to background noise. Specifically, such cable is specifically susceptible to background noise generated by external sources, such as adjacent lines, dimmer switches, AC and DC  
10 motors and modems.

## SUMMARY OF THE INVENTION

According to the invention, there is provided a method of automatically detecting a peak level of a signal propagated on a carrier  
5 medium. A traversal of a noise threshold level, determined relative to a noise signal, by a data signal is detected. A determination of whether a data signal traverses a peak level within a predetermined time interval after the detection of the traversal of the noise threshold level is performed. The peak level is varied in accordance with the  
10 determination of whether the data signal traversed the peak level.

Other features of the present invention will be apparent from the accompanying drawings and from the detailed description that follows.

## BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not limitation, in the figures of the accompanying drawings in which like references indicate similar elements.

5

**Figure 1** illustrates a Local Area Network (LAN) within which the present invention may be implemented.

**Figure 2** is a diagrammatic representation of a mapping of  
10 networking device components to the OSI model.

**Figure 3** is a diagrammatic representation of a MAC-to-PHY interface according to the present invention.

15 **Figure 4** is a diagrammatic representation of a frame constructed according to the teachings of the present invention.

**Figure 5** is a block diagram showing the relationship between the MAC frame, and a frame constructed according to the teachings of the  
20 present invention.

Figure 6 is a block diagram illustrating a network station, including a modem, according to an exemplary embodiment of the present invention.

5        **Figure 7** is a block diagram illustrating further components that constitute the modem, according to an exemplary embodiment of the present invention.

10        **Figure 8** is a diagrammatic representation of an exemplary header portion of a data transmission constructed according to the teachings of the present invention.

15        **Figure 9** is a diagrammatic representation of an exemplary data portion of a data transmission constructed according to the teachings of the present invention.

**Figure 10** illustrates an exemplary delimiter in the form of a pulse doublet.

20        **Figure 11** illustrates an exemplary delimiter in the form of a multi-cycle waveform.

Figure 12 illustrates an exemplary spectral mask to which the output power spectrum of a transmitter within the modem, illustrated in Figure 7, may conform.

5        **Figure 13** is a diagrammatic representation of a synchronization interval, and two access identifies symbols, as may be propagated from a transmitter, according to an exemplary embodiment of the present invention.

10        **Figure 14** is a signal diagram illustrating an exemplary multi-cycle waveform, a derived envelope signal, and exemplary peak threshold, data threshold, and noise threshold levels as determined for detection of the envelope signal.

15        **Figure 15** is a block diagram illustrating further components detail for an exemplary receiver, included within the modem illustrated in Figure 7.

20        **Figure 16** is a block diagram illustrating further details regarding pulse detect logic and threshold logic, according to an exemplary embodiment of the present invention, that may be included within the receiver illustrated in Figure 15.

**Figure 17** is a state diagram illustrating a state set for a "squellch" state machine that may be implemented within a noise follower circuit, as illustrated in Figure 16, according to an exemplary embodiment of the present invention.

5

**Figure 18** is a flowchart illustrating a method, according to an exemplary embodiment of the present invention, of dynamically varying the noise sensitivity of a receiver.

10

**Figure 19** is a timing diagram illustrating an exemplary envelope signal and an exemplary noise comparator signal in two scenarios.

**Figure 20** is a timing diagram illustrating an exemplary envelope signal, noise comparator signal and D.C. envelope detect signal, according to one embodiment of the present invention.

15

**Figure 21** is a timing diagram illustrating an exemplary envelope signal, noise comparator signal, receive carrier sense signal, and continuous carrier detect signal, as generated according to one embodiment of the present invention.

20

**Figure 22** is a timing diagram illustrating an exemplary envelope signal and an exemplary noise comparative signal that illustrate a



problem which may occur when a high noise event is followed by a packet transmission having a low signal amplitude.

5       **Figure 23** is a timing diagram illustrating an exemplary envelope signal, noise trigger signal, peak trigger signal, miss signal, and hit signal, as generated according to one embodiment of the present invention.

10       **Figure 24** is a flowchart illustrating a method, according to an exemplary embodiment of the present invention, of raising and lowering a peak threshold level.

15       **Figure 25** is a state diagram illustrating a state set, according to one exemplary embodiment of the present invention, that describes how peak search logic modifies an initial peak threshold value as a result of the assertion of the hit and miss signals illustrated in Figure 23.

**Figure 26** is a flowchart illustrating a method, according to an exemplary embodiment of the present invention, by which a mode of operation for the calculation of a data threshold level is determined.

## DETAILED DESCRIPTION

A method and apparatus for automatically determining a peak voltage level for a data signal propagated on a carrier medium are described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be evident, however, to one skilled in the art that the present invention may be practiced without these specific details.

### **Apparatus-Overview**

10       **Figures 1** illustrates a Local Area Network (LAN) 10 that may be implemented utilizing arbitrary topology Plain Old Telephone Service (POTS) wiring 14, such as Unshielded Twisted-Pair (UTP) wiring, within a building 16 that may comprise a residential dwelling, apartment complex, hotel, multi-unit dwelling complex, small business office, or  
15       any other building within which a POTS wiring network exists. The wiring 14 fans out from a central connection point 18, through which the wiring 14 is connected to a central office, to numerous phone jacks 20 located throughout the building 16. A number of LAN devices (also termed "stations" or "nodes"), such as computers 22, modems 24 or  
20       printers 26 may be coupled to the wiring 14 via the jacks 20. Regular telephones 28 are also shown to be coupled to the wiring 14 via respective jacks 20. A number of branches of the wiring 14 may furthermore terminate in jacks 20 into which LAN devices or telephones are not

connected (i.e., open stubs), and are accordingly unterminated. As will be appreciated from **Figure 1**, the wiring 14 interconnecting the LAN devices has an irregular topology, and includes a number of unterminated branches. Impedance mismatches, the arbitrary topology, and the lack of  
5 termination within the LAN 10 results in the reflection of signals and unpredictable frequency responses within the LAN 10. Further, the star configuration illustrated in **Figure 1** serves to attenuate the levels of signal features whose duration is short compared to the propagation delay of the wire branches.

10 **Figure 2** illustrates two exemplary networking devices 17 and 19, which are coupled via a wire pair 23 to a wiring network 25. Each of the networking devices 17 and 19 includes a physical layer device (PHY) 1022 or 1028, a media access controller (MAC) 1024 or 1030, and a logical link controller (LLC) 1026 or 1032. The various layers comprising the Open  
15 System Interconnect (OSI) networking model are illustrated at 21, and the PHY 1022 is shown to implement the physical layer 1020a while the MAC 1024 and the LLC 1026 are shown to implement the data link layer 1020b. In one embodiment, the present invention may reside primarily in the PHY 1022, and may adapt frames received from the MAC 1024 into a form  
20 suitable for reliable transmission over the wiring network 25 via the wire pair 23.

**Figure 3** provides further details regarding an exemplary embodiment of the MAC to PHY interface. Specifically, the interface

includes a receive data signal 1040, which comprises data propagated to the MAC 1024 from the PHY 1022. A receive clock signal 1042 is utilized to clock the receive data signal 1040. A transmit data signal 1044 comprises data propagated from the MAC 1024 to the PHY 1022. A  
5 transmit clock signal 1046 is utilized to clock the transmit data signal 1044. A transmit enable signal 1048 comprises a request from the MAC 1024 to the PHY 1022 to begin the transmission of data to the PHY 1022. A carrier sense signal 1050 indicates to the MAC 1024 that the PHY 1022 is receiving a valid carrier signal from the wiring network 1018. Finally, a  
10 collision signal 1052 indicates that a collision was detected by the PHY 1022 on the wiring network 25.

While the MAC 1024 and the PHY 1022 are shown to be discrete components in **Figure 3**, it will readily be appreciated that the MAC 1024 and the PHY 1022 may be integrated into a single device, in which case  
15 the above described signals will not be required. However, the functionality embodied in these signals should still be provided by the PHY component of any such integrated device.

### Frame Structure-Overview

**Figure 4** illustrates an exemplary embodiment of a frame 99 (or  
20 data packet) constructed by the PHY 1022, according to the teachings of the present invention, for transmission over the wire pair 23 within a structure. The frame 99 comprises a header portion 100 and a data portion 102, each of the header and the data portions 100 and 102 in turn

comprising a series of symbols. Specifically, the header portion 100 includes a series of access identifier symbols 103, which may be of a fixed length. The data portion 102 comprises a series of data symbols 105, which are of a variable length and, in one exemplary embodiment, encode an IEEE 802.3 Ethernet data package received from the MAC.

Figure 5 provides an exemplary illustration of the transformation of a transmission 1070, comprising a Ethernet packet 1072 having a preamble and delimiter 1074 prefixed to the Ethernet packet 1072, into a transmission 1090 for propagation over the wire pair 23. Specifically, the present invention proposes a method and apparatus whereby the PHY 1028 receives an Ethernet MAC frame (i.e., the transmission 1070) from the MAC 1024, strips away the eight octets comprising the preamble and delimiter 1074, and then replaces the preamble and delimiter 1074 with a header portion 100. Both the header portion 100 and the Ethernet packet 1072 are encoded as specific pulse sequences, as will be described in further detail below. In one exemplary embodiment, the step of transforming the transmission 1070 into the transmission 1090 comprises stretching the preamble and delimiter 1074 by introducing a "stretched clock" portion 1086 into the preamble to generate the intermediate header 1084, which comprises the contents of the delimiter and preamble 1074, but has a time duration of the header portion 100.

## Modem

085710-091598

Turning now to **Figure 6**, there is illustrated an exemplary network station 30 coupled to a LAN via uncharacterized, arbitrary topology wiring, such as the POTS wiring 14 illustrated in **Figure 1**. While the network station 30 is described below as implementing the Ethernet protocol, it will be appreciated that the teachings of the present invention can also be applied to other well-known networking protocols. Merely for example, the teachings of the present invention could be extended to any network implementing BISYNC or HDLC framing, Collision Detecting, Collision Avoiding, TDMA, and Polled or Token Passing access methods.

The network station 30 has a network device 36 and a physical-layer device (or PHY) in the form of an adapter 35. The adapter 35 includes a modem 32 that serves to adapt the transmissions of a serial communications controller 34 of the network device 36 (e.g., a computer) for propagation over the arbitrary POTS wiring 14.

The serial communications controller 34 may be an Ethernet communications controller operating according to the IEEE 802.3 standard High-level Data Link Control (HDLC) serial controller. In an alternative embodiment, the serial communications controller 34 may simply be a serial data interface or microprocessor, and the modem 32 may incorporate a Media Access Controller (MAC) that interfaces a data stream from the controller 34 to circuitry within the modem 32 via an

interface such as the General Purpose Serial Interface (GPSI) 60 described below.

The modem 32 provides medium interface, signal encoding and decoding (ENDEC), clock recovery and collision detection functions. To  
5 accommodate a variable bit rate encoding/decoding scheme, and to control the flow the data between the network device 36 and the wiring 14, the modem 32 controls the clocking of data to and from the serial communications controller 34. The modem 32 may further support Carrier Sense Multiple Access/Collision Detection (CSMA/CD) based  
10 Media Access Control (MAC) layers and accordingly may provide carrier detect and collision indication signals. The modem 32 is furthermore shown to be coupled to a microprocessor system 38 of the network device 36 via a management interface, which allows the modem 32 to be controlled by software executing within the microprocessor system 38.  
15 The adapter 35 further incorporates a coupler module 40 by which the network station 30 is coupled to the jack 20 and POTS wiring 14. The coupler module 40 includes a pair of RJ-11 ports through which the wall jack 20 and the telephone 28 are coupled to the network device 36. The coupler module 40 is furthermore connected to the modem 32 via a  
20 transmit/receive connection 42.

Referring now to the network device 36, the serial communications controller 34 is typically responsible for Link-Layer protocol functions, such as framing, error detection, address recognition

085710.P036

and media access. The microprocessor system 38 is responsible for the execution of software that controls the controller 34 and the modem 32. The microprocessor system 38 is furthermore shown to be coupled to the serial communications controller 34 via a data bus 44, and the

5 communications controller 34 is similarly coupled to the modem 32.

The adapter 35 thus allows a network device 36 including a communications controller 34, such as for example an Ethernet controller, to be coupled to a LAN implemented using POTS wiring 14, and serves to encode header and data information transmitted from the

10 network device 36 onto the LAN into a format suitable for propagation over the wiring 14. Similarly, the adapter 35 decodes signals received via the POTS wiring 14 into a format suitable for reception by the communications controller 34. While the adapter 35 is shown in **Figure 6** to reside outside the network device 36, it will be appreciated that the

15 adapter 35 could in fact be incorporated within the network device 36 as, for example, part of a Network Interface Card (NIC). Alternatively, the adapter 35 may comprise a stand-alone unit that is coupled between a serial port of the network device 36 and the wall jack 20.

**Figure 7** is a block diagram providing a more detailed view of the

20 components of the modem 32. Specifically, the modem 32 comprises an encoder/decoder (ENDEC) 46 that is responsible for encoding a bit sequence as a symbol for transmission over a carrier medium, such as the POTS wiring 14. Similarly, the ENDEC 46 is responsible for decoding a



symbol received from the wiring 14 to generate a bit sequence. The ENDEC 46 furthermore supplies both transmit and receive clocks to the communications controller 34 or, in an alternative embodiment, to a MAC within the modem 32. Overall operation of the ENDEC 46 is under  
5 the control of an access sequencer 48. The access sequencer 48 selects a general mode of operation for the ENDEC 46 and for the modem 32 as a whole, and controls the flow of data between the modem's subsystems and interfaces. The access sequencer 48 is also responsible for implementing collision detection within the modem 32.

10 The modem 32 further includes receiver/transmitter circuitry 50, which is responsible for the reception and transmission of (1) access identifier symbols 103 that encode unique identifiers for use in collision detection and (2) data symbols 105 that encode bit sequences. In one embodiment of the invention, such access identifier and data symbols 103  
15 and 105 may be delimited by, or include, electrical pulses. A transmitter 50a within the circuitry 50 receives symbol and, in one exemplary embodiment, polarity information that is synchronized to a master clock from the ENDEC 46. In one embodiment, the data symbol information received at the transmitter 50a from the ENDEC 46 represents the  
20 variable-duration (or encoding) portion of a composite symbol to be propagated from the transmitter 50a. The transmitter 50a appends a fixed-duration (or buffer) portion to the variable-duration portion to generate the composite symbol. In one embodiment of the present

invention, the transmitter 50a then generates data symbol delimiters, which determine the time duration and polarity of the composite symbol, in the form of pulse waveforms. In one embodiment, each pulse waveforms comprises a pulse "doublet" while, in a second embodiment, each pulse wherefore comprises a multi-cycle waveform.

Each pulse doublet may include first and second portions of approximately equal duration and opposite polarity, and have a fixed pulse width. The polarity of each pulse doublet is determined by information received from the ENDEC 46. As each pulse doublet comprises two equal portions of equal duration and opposite polarity, no Direct Current (DC) component is generated on the wiring 14 when the pulse doublet is transmitted thereon. Compliance with FCC Part 68 requires that the voltage levels of each pulse doublet be substantially less than would be generated when driving the coupler module 40 at Transistor-Transistor Logic (TTL) levels. A reduction in the voltage levels of each pulse may be accomplished by incorporating a series of resistors, or step-down windings, in a coupling transformer within the circuitry 50. Each pulse doublet may be a single cycle of a 2 MHz sine wave.

The circuitry 50 also includes a receiver 50b, which includes gain elements, comparators and digital control circuitry. The receiver 50b produces an output pulse, coincident with a predetermined point, or voltage, of a complex waveform received by the wall jack 20 off the

wiring 14. The complex waveform may be highly attenuated and distorted as a result of being propagated through a residential POTS wiring 14, and reflections can result in a peak amplitude occurring sometime after arrival of the incident energy. Without line terminations  
5 and with complex topologies, pulse energy may continue for many microseconds, decaying gradually. The receiver 50b further includes an Analog Front End (AFE) circuit 50c and a Phase-Lock Loop (PLL) circuit 50d. The AFE 50c shapes transmit pulses, and finds and indicates to the receiver 50b the time (temporal) location of a predetermined point or  
10 threshold voltage (e.g., the incident peak) of a received waveform for the purposes of identifying the access identifier and data symbols 103 and 105. Further details regarding the detection threshold voltages used for the identification of the access identifier and data symbols are provided below.

15 Referring again to **Figure 7**, the modem 32 also includes system interfaces in the form of a General Purpose Serial Interface (GPSI) 60 and a management interface 62. The GPSI 60 allows the modem 32 to control the clocking of data into and out of the communications controller 34.

A management interface 62 is a simple 4-wire serial interface for  
20 setting and reading management configuration information. A microprocessor within the communications controller 34 uses the interface 62 to establish the speed of operation, and to set the access sequencer's 48 mode of operation. In one embodiment the

communications controller 34 is an Ethernet controller, and the management parameters for the modem 32 are stored in a serial EEPROM, and loaded automatically every time the controller 34 is reset or initialized.

5

### **Frame Structure -**

#### **Access Identifier Symbols**

A collision detection protocol is implemented over a network 10, such as that illustrated in **Figure 1**, which may have an arbitrary topology and unterminated branches. In one exemplary embodiment, a network 10 implements the Ethernet protocol and thus a CSMA/CD mechanism is invoked for every transmission over the network 10. In such a network 10, each networks station 30 includes an Ethernet communications controller 34, which provides the CSMA/CD mechanism. However, as detailed above, an arbitrary topology renders traditional collision detection methodologies unreliable. Specifically, the occurrence of reflections and noise on a network using POTS wiring as a carrier medium render traditional "carrier sense" methodologies and techniques unreliable. Accordingly, the present invention proposes that the adapter 35, as shown in **Figure 6**, provide a "front-end" for the communications controller 34 which allows the detection of collisions in a manner which compensates for the unpredictability and arbitrary nature of a carrier medium.

Merely for example, assuming that a maximum node separation of 500 feet is specified in the network 10, a "slot time" of no more than 2 microseconds may exist. The term "slot time" comprises twice the propagation delay between nodes (i.e., "round-trip" propagation delay), and indicates the maximum amount of time required by a node to capture the carrier medium. In order to facilitate the detection of signals transmitted by multiple stations within such a slot time, the present invention proposes assigning a unique access identifier, for example in the form of an eight-bit binary number, to each station. This access identifier is a unique identifier, and is transmitted from each station as part of the header portion 100 prior to the transmission of the data portion 102.

The generation and transmission of a frame 99, such as that illustrated in **Figure 4**, typically commences when the communication controller 34 indicates to the modem 32 a desire to transmit by raising a transmission enable (TENA) signal. If the carrier medium is detected as being available (e.g., no transmitted pulses from other nodes have been seen on the POTS wiring 14 for N microseconds), the access sequencer 48 causes the transmitter 50a to transmit the header portion 100, which includes the access identifier unique to the network station 10. The access identifier is substituted for a portion of a preamble transmission, which is conventionally transmitted from an Ethernet MAC controller at the beginning of a packet, as illustrated in **Figure 5**.

09153310-091598  
865160-0785760

For the purposes of the specification, it is convenient to define a time unit in terms of which time intervals and temporal locations may be expressed. To this end, a time unit arbitrary termed a TIC is defined as comprising 0.1167 microseconds.

5       As shown in **Figure 8**, in one exemplary embodiment, the header portion 100 comprises a synchronization interval 101, at least four (4) access identifier symbols 103, and a silence interval 107. The synchronization interval 101 is of a fixed length, and is bounded by two electrical pulses which are 128 TICs (e.g. 14.93 microseconds) apart. The  
10   four access identifier symbols 103 are of equal duration (e.g., 128 TICs) and each comprise a buffer or "dead time" portion 106 and an identifier portion 104. Each buffer portion 106 is twice the duration of an Intersymbol Blanking Interval (ISBI) which comprises 20 TICs. The ISBI is further defined and explained in co-pending U.S. patent application  
15   Serial No. 08/925,205, entitled "METHOD AND APPARATUS FOR ENCODING AND DECODING A BIT SEQUENCE FOR TRANSMISSION OVER POTS WIRING", filed September 8, 1997, which is incorporated by reference into the present specification. Each buffer portion 106 may conveniently be termed an Access Identifier (AID) blanking interval. The  
20   respective buffer portions 106 are required to separate the pulses included within the identifier portions 104 by a predetermined time duration so as to allow for proper detection thereof by a receiver 50b.

00153810-091598  
BEST COPY AVAILABLE

Two bits are encoded in the temporal location of a pulse within each identifier portion 104. For example, a pulse transmitted in the 66th TIC of a symbol 103 encodes the bit sequence "00". Similarly, the bit sequences "01", "10", and "11", are encoded by pulses transmitted in the 86th, 106th and 126th TICs within a symbol 103. For example, the identifier portion 104 of the first access identifier symbol 103 includes a pulse transmitted in the 106th TIC of the symbol 103b, and thus encodes the bit sequence "10".

The silence interval 107 may also be 128 TICs in duration, and is used for the detection of a "jam" pattern, and it is during this silence interval 107 that all receivers 50b may attain maximum sensitivity.

As described in co-pending U.S. patent application Serial No. 08/925,043 entitled "METHOD AND APPARATUS FOR DETECTING COLLISIONS ON A NETWORK, filed September 8, 1997, which is incorporated by reference into the present specification, the ENDEC circuitry 46 may utilize the header portions 100 of received frames 99 to detect collisions over a network implemented on POTS wiring 14. In summary, by comparing the temporal locations of pulses of a received header portion 100 with the temporal locations of pulses within a transmitted header portion 100 unique to a respective network device, and detecting any variations between the temporal locations of corresponding pulses within such header portions 100, the modem 32 is

able to detect collisions in the manner detailed in the above identified co-  
pending U.S. patent applications.

09153810-091598  
BEST COPY AVAILABLE



085710.P036

## Frame Structure -

### Data Symbols

In seeking to address the problem of unreliable transmission over a network having an arbitrary topology and unterminated nature (e.g., a POTS wiring network 25), the present invention teaches encoding data (e.g., an Ethernet packet) as a sequence of data symbols 105 that together constitute the data portion 102 of a frame 99, as illustrated in **Figure 4**. In one embodiment, each data symbol 105 is distinguished by having a specific time duration, this time duration being determined by the time interval between the generation (or receipt) of first and second delimiters in the exemplary form of pulse waveforms. More specifically, data is encoded in the precise time between the generation (or arrival) of short duration pulses that are widely separated in time.

As shown in **Figure 9**, in one exemplary embodiment, the data portion 102 includes a sequence of data symbols 90 that may each be of varying duration, and which are defined by a sequence of pulses 92 transmitted from a receiver 50b over the POTS wiring 14. Each data symbol 90 has a predetermined duration that is indicative of a predetermined bit sequence. A minimum time duration separates each of the pulses 92 from a preceding pulse, so as to allow reflections resulting from the pulse 92 on the POTS wiring 14 to decay below a predetermined threshold before transmission of a subsequent pulse 92. This minimum time duration is embodied in a fixed-duration blanking

interval 94 that proceeds the generation of each pulse 92. Following the blanking interval 94, each data symbol 90 then includes a variable-duration symbol portion 96 that encodes the predetermined bit sequence. Each symbol portion 96 is demarcated by (1) the end of a preceding  
5 banking interval 94 and (2) a pulse 92 that is generated in one of twenty-four temporal locations, each of the twenty-four temporal locations being indicative of a predetermined bit sequence, as described in co-pending U.S. patent application Serial No. 08/925,205. For example, the second pulse 92 shown in **Figure 9** may be located at a 7th location, while the  
10 third pulse 92 may be located at the 12th location. In this case, the symbol portion 96 of the first data symbol 90 will be shorter in time duration than the symbol portion 96 of the subsequent data symbol 90.

### Pulse Shape

The pulses included within the access identifier symbols 103, and  
15 which comprises the delimiters of the data symbols 105, may comprises pulse doublets, as shown at 300 in **Figure 10**. The pulse doublets 300 each include two equal portions of equal duration and opposite polarity, so that no direct current (DC) component is generated on a wiring pair 23 when the pulse doublet is transmitted thereon.

20 In an alternative embodiment of the present invention, the pulse may take the form illustrated in **Figure 11**. Specifically, each pulse may comprise a multi-cycle waveform 302. The waveform 302 may be a 7.5 MHz waveform that is propagated from the transmitter 50a, and may

include four cycles per pulse. In one embodiment, the waveform 302 is formed from an integer number of cycles of a square wave having a frequency of 7.5 MHz that has been filtered using a bandpass filter, such as for example a five pole Butterworth filter having a passband of between 5.5 and 9.5 MHz. **Figure 12** illustrates a spectral mask 306 to which the output power spectrum of a transmitter 50a may conform. The power level requirements specified by the spectral mask are below 1.1 MHz to allow the modem 32 to interoperate with Analog Digital Subscriber Line (ADSL) modems.

**Figure 13** is a diagrammatic representation of an exemplary synchronization interval 101, and two exemplary access identifier symbols 103, as propagated from the transmitter 50a. Specifically, it will be noted that the pulses which demarcate the beginning and the end of the synchronization interval 101 at 2010 and 2012 comprise multi-cycle waveforms, corresponding in shape to the multi-cycle waveform 302 shown in **Figure 11**. Similarly, the pulses which demarcate temporal locations within the access identifier symbols 103 comprise multi-cycle waveforms. Each of the multi-cycle waveforms comprising a pulse have, in one exemplary embodiment, a frequency of above 1.1 MHz, and more specifically, a frequency of between 5.5 and 9.5 MHz, and most specifically, a frequency of approximately 7.5 MHz. A temporal location within a symbol 103 is marked by the transcendence of a predetermined noise threshold voltage by a first cycle of the multi-cycle waveform. Pulses in

the form of the multi-cycle waveforms illustrated in **Figure 13** are advantageous in that transmissions from the receiver 58a, for example, at 7.5 MHz may inter-operate with modems implementing an xDSL protocol, which typically operate at below 1.1 MHz.

5                                    **Noise, Data and Peak Threshold Levels**

As discussed above with reference to **Figure 4**, a frame 99, constructed by the PHY 1022, includes a header portion 100 and a data portion 102. The header portion 100 is utilized for the purposes of, inter alia, synchronization and collision detection, while the data portion 102 is utilized for the actual transmission of data. In order to allow for the detection of collisions with packets emanating from distant nodes on the wiring network 25, it is desirable that the sensitivity of the receiver 50b be maintained at a high level during collision detection (i.e. during reception of a header portion 100). On the other hand, in order to reject noise impulses and the decaying remnants of a previous pulse, it is desirable that the sensitivity of the receiver 50b be at a lower level during reception of a data portion 102. To this end, the present invention proposes implementing three (3) threshold levels within the receiver 50b that allow the receiver 50b to utilize different levels of sensitivity during the detection of the header portion 100 and during the detection of the data portion 102. Referring now to **Figure 14**, there are illustrated two exemplary signals that have been superimposed on each other, namely a received multi-cycle waveform 140 and a unipolar pulse waveform 142

(hereafter referred to as an envelope signal 142) that comprises a detected envelope of the waveform 140. Three threshold levels are also shown to be traversed by the envelope signal 142, namely a noise threshold level 144, a data threshold level 146 and a peak threshold level 148. The noise threshold level 144 is shown to be ideally situated just above a measured noise floor, represented by noise pulses 150 and, as will be explained in greater detail below, may be adjusted to track an ambient noise level.

### The Receiver

Figure 15 is a block diagram illustrating further component detail of the receiver 50b, according to an exemplary embodiment of the present invention. The receiver 50b performs the following functions to decode a received waveform:

1. Rejects noise pulses.
2. Rejects decaying signal remnants of current or previous pulses.
- 15 3. Selects a distinctive feature of the waveform for timing measurement.
4. Accurately measures the time delay from the previous pulse.
5. Maximizes noise rejection during data symbol intervals.

By designing the electrical characteristics of the receiver 50b to bandlimit the incoming waveform to between 5.5 and 9.5 MHz, band noise and harmonic power may be effectively rejected. The receiver 50b is shown to include the Analog Front End (AFE) 50a, which in turn comprises a socket 160 for receiving a jack coupling the receiver 50b to

the POTS wiring network 25. The multi-cycle waveform 140, as illustrated in **Figure 14**, is shown to be propagated from the socket 160 to a Butterworth filter 162, which implements the spectral mask shown in **Figure 12**, and accordingly limits the bandwidth of the received signal to between 5.5 and 9.5 MHz. The filtered signal is then propagated from the Butterworth filter 162 to a differential amplifier 164, and then onto an envelope detector 166 that outputs the envelope signal 142 illustrated in **Figure 14**. The envelope signal is then fed through a unipolar operational amplifier (opamp) 168, from where it is delivered to pulse detect logic 170, which is illustrated to send and receive signals from "squelch" or threshold logic 172. Specifically, the threshold logic 172 propagates analog signals to the pulse detect logic 170 that indicates the noise threshold level 144, the data threshold level 136, and the peak threshold level 148. The pulse detect logic 170 then outputs a sequence of digital pulses to the decoder 46b, this sequence of digital pulses encoding both the access identifiers and data.

**Figure 16** is a block diagram providing further detail regarding the pulse detect logic 170 and the threshold logic 172, according to an exemplary embodiment after the present invention. The threshold logic 172 is shown to be implemented utilizing a Field Programmable Gate Array (FPGA) 173. It will however be appreciated that, in alternate embodiments, the functionality described below could be implemented in hardware, software, or any combination thereof. In the case where any

of the functionality and logic required to implement the invention as described below, or as embodied within the claims, is implemented in software, the sequence of instructions embodying the functionality, and that are executed by a machine (e.g., a general purpose microprocessor or  
5 microcontroller), may be stored utilizing any machine-readable medium that is accessible by the machine. Accordingly, for the purposes of the present specification, the term "machine-readable medium" shall be taken to include, but not be limited to, solid-state memories, magnetic memories, optical memories, and carrier wave signals.

10 The pulse detect logic 170 is shown to include three comparators, namely a peak comparator 180, a noise comparator 182 and a data comparator 184. Each comparator is illustrated as receiving two inputs, namely the amplified envelope signal 142 on line 186 and a respective threshold level from the threshold logic 172. The peak comparator 180 is  
15 shown to receive the peak threshold level 148 on line 188, the noise comparator 182 is shown to receive the noise threshold level 144 on line 190, and the data comparator 184 is shown to receive the data threshold level 146 on line 192. Each of the comparators compares the amplified envelope signal 142 to a respective threshold level, and outputs a square  
20 pulse on the detection of a transcendence, or traversal, of the respective threshold level by the amplified envelope signal 142. Specifically, the peak comparator 180 outputs a peak comparator signal 181, the noise comparator 182 outputs a noise comparator signal 183, and the data

comparator 184 outputs a data comparator signal 185, each of these signals comprising a sequence of spaced pulses in the form of square waves.

The threshold logic 172 includes a noise follower 194, that implements a "squellch" state machine 196 that is further shown to include an event counter 198, a one (1) second timer 200 and a fifteen (15) millisecond timer 202. The timers 200 and 202 may be implemented utilizing a single timer in conjunction with a chain of dividers, and the noise follower 194 may accordingly include any number of timers. The noise follower 194 is further shown to receive three primary inputs, namely a signal quality error signal 204, a continuous carrier detect signal 206 and a DC envelope detect signal 208. Both the signal quality error signal 204 and the DC envelope detect signal 208 are, in one exemplary embodiment, fed back to the noise follower 194 from a MAC controller 34, which is shown to receive the noise comparator signal 183 as an input. The noise comparator signal 183 is utilized by the MAC controller 34 to generate the signal quality error signal 204, and also to decode and determine an access identifier of a remote, transmitting node for the purposes of collision detection. The MAC controller 34 is also shown to receive a receive carrier sense signal 205, which is utilized in the manner described below to generate the continuous carrier detect signal 206. The utilization of each of the signals by the noise follower 194 will be described in further detail below. The noise follower 194 furthermore outputs seven-bit sequences, indicative of the noise threshold level 144,



to a digital-to-analog converter 210, that outputs the noise threshold level 144 as an analog signal.

The threshold logic 172 further includes peak hit logic 212 and peak search logic 214. The peak hit logic 212 is coupled to receive a noise trigger signal 222 and a peak trigger signal 224 from synchronization circuitry 226 that generates the trigger signals 222 and 224 as rising edge synchronized derivatives of the noise comparator and peak comparator signals 183 and 181. The peak hit logic 212 outputs respective hit and miss signals 228 and 230 to the peak search logic 214. The peak search logic 214 in turn propagates seven-bit sequences to a digital-to-analog converter 216, that outputs an analog signal comprising the peak threshold level 148.

Finally, averaging logic 218 receives the seven-bit sequences outputted from the noise follower 194 and the peak search logic 214, and performs an averaging operation on these outputs to generate a unique seven-bit sequence output that is similarly propagated to a further digital-to-analog converter 220, the outputs of the converter 220 comprising the data threshold level 146.

### Noise Threshold Level Algorithm/State Machine

**Figure 17** is a state diagram illustrating a state set 240 for the "squellch" state machine 196, according to an exemplary embodiment of the present invention. From a reset state 242, the state machine 196 enters a clear state 244, wherein the event counter 198, the fifteen

millisecond timer 202 (i.e., timer 1) and the one second timer 200 (i.e., timer 2) are all cleared to zero (0). From the clear state 244, the state machine 196 automatically progresses to the resting state 246, where the state machine 196 resides until one of a number of events occur to

5 indicate that the noise threshold level 144 needs to be raised or lowered. Specifically, the state machine 196 may progress to a raise state 254, as indicated by arrow 248, to a down state 256, as indicated by arrow 250, or back to the clear state 244, as indicated by the arrow 252. The transitions from the resting state 246 are furthermore prioritized, with the transition

10 to the raise state 254 having a highest priority, the transition to the down state 256 having an intermediate priority, and the transition to the clear state 244 having a lowest priority. A number of transition events determine to which state the state machine 196 transitions from the resting state 246, these transition events being indicated in respective

15 blocks 260-268 shown connected to the various transition arrows 248, 250 and 252.

Dealing firstly with the transition events 260, 266 and 268 that trigger a transition 248 from the resting state 246 to the raise state 254, an exemplary embodiment after the transition event 260 is illustrated in the

20 flowchart shown in **Figure 18**. Specifically, the transition event 260 commences at step 280, and proceeds to step 282, wherein the event counter 198, and the timers 200 and 202 are each cleared, this step being performed in the clear state 244 shown in **Figure 17**. At decision box 284,

a determination is made as to whether the timer 202 has recorded the expiration of fifteen millisecond. If so, a determination is then made at decision box 286 as to whether a number of noise events, as counted by the event counter 198, exceeds a predetermined maximum number  
5 (MAX \_ EVENTS) of noise events. In one exemplary embodiment, the predetermined maximum number (MAX \_ EVENTS) is equal to fifteen (15) noise events. In this regard, the event counter 198 is increment by, inter alia, each assertion of the signal quality error signal 204 by the MAC controller 34.

10 The MAC controller 34 is shown to receive the noise comparator signal 183, which is utilized by error logic 35 and a timer 37 within the MAC controller 34 to generate the signal quality error signal 204. The generation of the signal quality error signal 204 by the error logic 35 will now briefly be described with reference to **Figure 19**. **Figure 19** is a timing  
15 diagram showing an exemplary envelope signal 142 and a noise comparator signal 183 for two scenarios. Specifically, **Figure 19** illustrates a first scenario 288 in which a noise event will not be registered by the error logic 35, and a second scenario 290 in which a noise event will be registered by the error logic 35. As described above with reference to  
20 **Figure 8**, the header portion 100 of a frame 99 includes a synchronization interval 101 defined by two pulses that are spaced by a fixed and predetermined time interval. In the first illustrated scenario 288, an envelope signal 142 derived from a received header portion 100 is shown

to include a first pair of pulses 292 and 294 that defined the synchronization interval 101 of the header portion 100. The traversal of a noise threshold level 144 is detected by the noise comparator 182, which is then shown to generate a noise comparator signal 183, including pulses 5 296 and 298, in response to the respective traversals of the noise threshold level 144 by the pulses 292 and 294. The noise comparator signal 183 is then propagated to the MAC controller 34, which measures the time interval between the reception of the pulses 296 and 298 utilizing the timer 37. The error logic 35 then makes a determination as 10 to whether the measured time interval between the arrival of the pulses 296 and 298 corresponds to a synchronization interval and if so, registers the beginning of a packet reception. This is the case for the pulses 296 and 298. The second scenario 290 illustrates a situation in which two noise pulses 300 and 302 are included within an envelope signal 142, and 15 similarly traverse the noise threshold 144 to generate noise comparator signal pulses 304 and 306. In this case, the error logic 35 will detect the absence of a second pulse at a predetermined synchronization interval after the detection of the pulse 304, recognize the pulse 304 as corresponding to a noise event and then assert the signal quality error 20 signal 204 responsive to the detection of the absence of the second pulse. It will be appreciated that, in a third scenario (not illustrated), should a noise pulse cause a noise comparator signal pulse to be generated prior to the expiration of the predetermined synchronization interval, the error

logic 35 will similarly recognize the premature noise comparator signal pulse as indicating a noise event, responsive to which the error logic 35 will then assert the signal quality error signal 204.

Returning now to the flowchart shown in **Figure 18** and the state diagram shown in **Figure 17**, should it be determined at decision box 286  
5 that the number of noise events counted by the event counter 198 within a fifteen millisecond time interval does exceed the predetermined maximum number of events (MAX \_ EVENTS), the noise threshold level 144 is raised at step 308 by transitioning the state machine 196 into  
10 the raise state 254. Prior to raising the noise threshold level 144, the state machine 196 determines that a noise ceiling will not be exceeded by the adjusted noise threshold level 144. The raising of the noise threshold level 144 is achieved by increment in the seven-bit output of the noise follower 194 by a predetermined amount. For example, the seven-bit  
15 output of the noise follower 194 may be incremented to achieve a 10 mV increase in the output of the digital-to-analog converter 210. Following the raising of the noise threshold level 144 at step 308, the new threshold level 144 is given the opportunity to settle at step 310. The settling interval is implemented so as to allow the increase in the threshold level  
20 144 to be propagated from the noise follower 194, through the digital-to-analog converter 210, and then to stabilize on the line 190. In one exemplary embodiment, the settling interval may be between 15-20

00153810-091598  
085710-0785760

milliseconds. From step 310, the event counter 198 and the timers 200 and 202 are again reset and cleared at step 282.

Returning to decision boxes 286 and 284, if it is determined at decision box 284 that the timer 202 has not registered fifteen milliseconds, or if it is determined at decision box 286 that the number of noise events counted by the event counter 198 does not exceed the predetermined maximum number of events, a determination is then made at decision box 311 whether the timer 200 as registered a one second time period. If not, a return is made to decision box 284. Alternatively, should the timer 200 register a one second interval, a further determination is made at decision box 312 whether the number of noise events registered by the event counter 918 is greater than, or equal to, a further predetermined number of noise events (RAISE \_ SECOND). In one exemplary embodiment, the further predetermined number of noise events (RAISE \_ SECOND) is four (4) noise events. If so, the noise threshold level 144 is raised at step 314, in the matter described above, whereafter the newly adjusted noise threshold level 144 is allowed to settle at step 310. The methodology discussed with respect to **Figure 18** thus far represents exemplary transition event 260.

Turning now to the transition event 262 that triggers the transition 250 from the resting state 246 to the down state 256, should it be determined at decision box 312 that the number of events registered by the event counter 198 is not greater than, or equal to, the further

predetermined maximum number of noise events (RAISE \_ SECOND), a further determination is then made at decision box 316 as to whether the number of noise events registered by the event counter 198 is equal to zero. If so, the state machine 196 then transitions from the resting state 246 to the down state 256, and the noise threshold level 144 is lowered at step 318 (e.g., by 10 mV) by decrementing the seven-bit output of the noise follower 194. The state machine 196 then transitions to the settle state 270, where the adjusted noise threshold level 144 is afforded the opportunity to settle at step 320. From the settle state 270, the state machine 196 transitions to the falling state 272 where a further determination is made at decision box 322 whether a further noise event is detected within a predetermined time interval. In one exemplary embodiment, a determination may be made as to whether a further noise event, represented by an assertion of the signal quality error signal 204, is detected within a 60 millisecond time interval. If no further noise event is detected, then the state machine 196 returns to the down state 256, where the noise threshold level 144 is again lowered by the predetermined increment. The falling state 272 allows the state machine 196 to rapidly decrease or "decay" the noise threshold level 144 (and accordingly increase the sensitivity of the receiver 50b) in the absence of any noise on the carrier medium. On the other hand, when in the falling state 272, should a noise event be detected within the 60 millisecond time interval at decision box 322, the state machine 196 transitions

directly to the raise state 254 where the noise threshold level 144 is raise at step 314 in the manner described above.

Turning now finally to the transition event 264 that triggers the transition from the resting state 246 to the cleared state 244, if it is  
5 determined at decision box 316, as illustrated in **Figure 18**, that the number of noise events registered by the event counter 198 is not equal to zero (e.g., the number of noise events registered by the event counter 198 is either 1,2 or 3) then the state machine 196 transitions from the resting state 246 to the clear state 244. In the clear state 244, the event counter 198,  
10 and timers 200 and 202, are reset and cleared at step 282. Accordingly, the state machine 196, by implementing the state set 240, seeks to maintain the noise threshold level 124 as a level at which between zero and three noise events are detected and registered per second.

It will also be noted that by comparing the number of noise events  
15 within a relatively short time interval (e.g., fifteen milliseconds) to a relatively large number of noise events (MAX \_ EVENTS), and raising the threshold level if the number of noise events within this short time interval exceeds the larger number of noise events, an accelerated increase in the noise threshold level 144 may be achieved in the event  
20 that a high occurrence of noise is present. In the event that less noise is present on the carrier medium, the rate at which the threshold level 144 is raised is slower, in that the threshold level 144 will only be raised if a



09153810-091598  
BEST COPY AVAILABLE

relatively small number of noise events (RAISE \_ SECOND) are detected within a relatively long time interval (e.g., one second).

The transition event 266, that may conveniently be termed a DC envelope detect transition, will now be described with reference to **Figure 20**. **Figure 20** illustrates three exemplary signals, namely the envelope signal 142, the noise comparator signal 183 and the DC envelope detect signal 208. In the event that a DC signal is generated on the carrier medium, for example by an AM radio carrier signal or the like, an envelope signal 142 as illustrated in **Figure 20** may be inputted to the noise comparator 182. In this case, as the envelope signal traverses the noise threshold level 144 in a rising direction, but may then maintain a roughly constant level (e.g., approximately 200 mV) and may not again fall below the noise threshold level 144. In this case, the output of the noise comparator 182 (i.e., the noise comparator signal 183) will not be de-asserted. As the MAC controller 34, and the error logic 35, detect signal quality errors responsive to the rising edge of the noise comparator signal 183, an observation of the signal quality error signal 204 will not indicate the presence of any noise events on the carrier medium. Accordingly, the state machine 196 will view the carrier medium as being quiet, and maintain the noise threshold level 144 at a current level (e.g., 50 mV), well below the noise floor (e.g., 200 mV) of the DC envelope signal 142. To address the above described limitation, the present invention proposes that the timer 37 time the interval for which the noise

09153810-091593  
085710-0785760

comparator signal 183 is asserted, and that the error logic 35 generate the DC envelope detect signal 208 pulse at predetermined time intervals during the assertion of the noise comparator signal 183. In one exemplary embodiment, the error logic 35 may output a pulse, as  
5 illustrated at 340, after each 30 millisecond time interval for which the noise comparator signal 183 is asserted. The event counter 198 is incremented by each pulse of the DC envelope detect signal 208, in the same way that it is incremented by a pulse of the signal quality error signal 204. Accordingly, each 30 millisecond time interval for which the  
10 noise comparator signal 183 is asserted is viewed as a noise event. Should the noise comparator signal 183 remain asserted as a result of a DC envelope signal 142, the event counter 198 will be incremented at regular intervals (e.g., 30 milliseconds), thus resulting in the noise threshold level 144 being raised until it exceeds the noise floor presented  
15 by the DC signal, at which stage the noise comparator signal 183 will be de-asserted.

The transition event 268, which may conveniently be termed a carrier sense violation transition, will now be described with reference to **Figure 21**. **Figure 21** again illustrates an exemplary envelope signal 142,  
20 an exemplary noise comparator signal 183, a receive carrier sense signal 205, and a continuous carrier detect signal 206. The exemplary envelope signal 142 is shown to comprise unipolar pulses 350 and 352 that are relatively close together (e.g., spaced by less than 8 microseconds). Such

closely spaced pulses may result from interference generated by, for example, a DC motor. As described above with reference to **Figure 19**, the synchronization interval 101 allows the MAC controller 34 to detect the beginning of a frame transmission if first and second pulses are spaced by a predetermined interval (e.g., 14.9 microseconds). In the absence of a second pulse at the 14.9 microsecond location after the detection of a first pulse, a noise event is registered. However, should a further noise pulse, such as that illustrated at 352, be received prior to the 14.9 microsecond location, the receive carrier sense signal 205 may not turn off as a result of the continually received signals. As shown in **Figure 16**, the receive carrier sense signal 205 is provided to the MAC controller 34, where the timer 37 times interval for which the receive carrier sense signal 205 is asserted. In the event that the receive carrier sense signal 205 remains asserted for greater than a predetermined time ( e.g., 30 microseconds), the continuous carrier detect signal 206 is asserted by the error logic 35 to generate a pulse 354. On detecting the pulse 354 of the continuous carrier detect signal 206, the state machine 196 will then transition from the resting state 246 to the raise state 254, where the noise threshold level 144 will be raised in the manner described above with reference to **Figure 18**. Accordingly, the transition event 268 differs from the transition events 260 and 266 in that such an event does not incremented the event counter 198, but causes the state machine 196 to transition immediately from the resting state 246 to the raise state 254.

085710.P036

A further embodiment of the present invention will now be described with reference to the timing diagram shown in **Figure 22**. A situation may occur in which a time interval of high noise levels is proceeded by a data transmission for which the signal amplitudes are relatively small relative to the amplitudes of the preceding noise signal. **Figure 22** illustrates such a situation wherein an envelope signal 142 comprises a sequence of large amplitude pulses (i.e., a high noise event) followed by a sequence of low amplitude pulses that constitutes a valid data transmission. A dynamically varying noise threshold level 124 is also shown to follow the noise floor, for example by the methodologies discussed above, following the ending of the high noise event. However, it may occur that the noise threshold level 144 drops to a level at which the data transmission is detected sometime after the actual beginning of the transmission of the data packet. In other words, the noise threshold level 144 drops into the middle of a data transmission. In such a case, a pulse of the envelope signal 142 representing the data transmission may be recognized by the error logic 35 within the MAC controller 34 as a noise event in view of the MAC controller's ignorance regarding the preceding pulses of the data transmission. This may in turn result in the noise threshold level 144 being raised again, as illustrated in **Figure 22**, to a level above which the pulses of the data transmission are detected. With a view to preventing the occurrence of such an event, one embodiment of the present invention proposes that the signal quality

error signal 204 only be asserted if a noise event is detected, and the receive carrier sense signal 205 has not been asserted for longer than a predetermined interval (e.g., 300 microseconds). Specifically, if the receive carrier sense signal 205 has been asserted for greater than the predetermined time interval, then a pulse of the noise comparator signal 183, that would otherwise have generated a pulse of the signal quality error signal 204, is not viewed as indicating a noise event. If the receive carrier sensor signal 205 has been asserted for greater than 300 microseconds, this indicates that a packet may be in the process of being received, and the noise threshold level 144 will not be raised. It will also be appreciated that, in the event that the lower amplitude signal is in fact noise, the receive carrier sensor signal 205 may remain asserted for longer than the 30 millisecond time interval specified above, in which case the continuous carrier detect signal 206 will be asserted to raise the noise threshold level 144.

### Peak Threshold Level Algorithm

A peak threshold level determination algorithm will now be described with reference to **Figures 16, 23, 24 and 25**. **Figure 23** is a timing diagram illustrating exemplary signals that may be generated within the receiver 50b, shown in **Figure 15**. Specifically, **Figure 23** illustrates an envelope signal 142, over which a noise threshold level 144, a data threshold level 146 and a peak threshold level 148 have been superimposed. The peak threshold level 148 is set to an initial high

value, and is then shown to incrementally descend until traversed by the envelope signal 142, after which it stabilizes at a level approximate with the peak level of the envelope signal 142. The methodology by which the peak level threshold 148 is lowered as shown in **Figure 23** will be

5 described below. **Figure 23** also illustrates an exemplary noise trigger signal 222 and an exemplary peak trigger signal 224. The signals 222 and 224 are clock synchronized derivatives of the noise comparator signal 183 and the peak comparator signal 181, and are generated by the synchronization circuitry 226, as illustrated in **Figure 16**. For the

10 purposes of explanation, the noise trigger signal 222 and the peak trigger signal 224 are shown to be derived directly from the envelope signal 122. However, as a result of the synchronization, it will be appreciated that the signals 222 and 226 are not aligned with transitions of the threshold levels, as shown in **Figure 23**. As stated above, the peak threshold level

15 148 may be set to an initial high level, at which it is not traversed by the envelope signal 142. In this case, the peak comparator signal 181 will not be asserted. However, it will be appreciated that the noise threshold level 144 is traversed by each pulse of the envelope signal 142, and will accordingly generate a series of pulses 360 of the noise trigger signal 222,

20 as illustrated. It will also be appreciated that, when the peak threshold level 148 eventually descends below the peak value of a pulse of the envelope signal 142, as illustrated at 362, the envelope signal pulse will traverse the peak threshold 148 at a time period slightly after the same

pulse has traversed the noise threshold 144, as a result of the slope of the rising edge of the pulse. The traversal of the peak threshold level 148 by an envelope signal pulse causes a pulse 364 of the peak trigger signal 224 to be generated, this pulse 364 being slightly delayed relative to a  
5 corresponding pulse 360 of the noise trigger signal 222. The present invention proposes utilizing this information to raise and lower the peak threshold level 148.

Referring now specifically to **Figure 24**, there is shown a flowchart illustrating a method 370, according to an exemplary embodiment of the  
10 present invention, of raising and lowering the peak threshold level 148. The method 370 commences at step 372, and then proceeds to step 374, where a determination is made by the peak hit logic 212 as to whether the noise trigger signal 222 has been asserted, as a result of a corresponding assertion of the noise comparator signal 183. If not, the method 370 loops  
15 in this state until a noise trigger signal assertion is detected. Following the detection of such an assertion, a determination is made by the peak hit logic 212 at decision box 376 regarding whether an assertion of the peak trigger signal 224 occurs within a predetermined time interval (e.g., a one microsecond). To this end, the peak hit logic 212 includes a timer  
20 for tracking the time interval following the detected assertion of the noise trigger signal 222. If an assertion of the peak trigger signal 224 is detected within the predetermined time interval, the hit signal 228 is asserted at step 378. Alternatively, if no assertion of the peak trigger

signal 224 is detected within the predetermined time interval, the miss signal 230 is asserted at step 380. The method 370 then loops back to decision box 374.

5       **Figure 25** is a state diagram showing a state set 390, according to one exemplary embodiment of the present invention, that describes how the peak search logic 214 modifies an initial peak threshold value, represented by a seven-bit sequence, as a result of the assertion of the hit or miss signals 228 or 230 by the peak hit logic 212. Specifically, the peak search logic 214 is shown to reside in an initial idle state, whereafter  
10       sequentially decreasing values are either added or subtracted to a preceding threshold value (from which the peak threshold level 148 is derived) dependent upon whether the hit signal 228 or the miss signal 230 is asserted.

#### **Data Threshold Level Algorithm**

15       As described above with reference to both, and as can be seen in **Figure 23**, the data threshold level 146 may, in one exemplary mode of operation of the receive the 50b, be located approximately mid-way between the peak threshold level 148 and the noise threshold level 144.

20       It is advantageous to locate the data threshold level 146 midway between the threshold level 144 and 148 as the pulses of the envelope signal 142 typically exhibited the greatest slope at 50 percent of the peak threshold level 148. Locating the data threshold level 146 at a level at which the envelope signal 142 exhibits the greatest slope results in the



transcendence of the data threshold level 146 by the envelope signal 142 being less susceptible to time-jitter that may be introduced by a given amount of amplitude noise. Nonetheless, it is also desirable that traversal of the data threshold 146 by noise signal is be reduced, and the data threshold level 146 should accordingly be raised when noise is detected on the carrier medium. Averaging the noise and peak threshold levels 144 and 148 provides a useful way to locate the data threshold level 146 in a voltage range where the envelope signal 142 may exhibited a steep slope, while also factoring in noise signal amplitude considerations.

10 In one embodiment, the calculation of the data threshold level 146 is performed by the averaging logic 218, illustrated in **Figure 16**, that receives the seven-bit output sequences of both the noise follower 194 and the peak search logic 214, performs an averaging operation utilizing these seven-bit output sequences, and generates a unique seven-bit output sequence to the digital-to-analog converter 220. The converter 220 in turn outputs an analog signal, derived from the seven-bit output sequence of the averaging logic 218, that constitutes the data threshold level 146.

20 **Figure 26** is a flowchart illustrating a method 400, according to an exemplary embodiment of the present invention, by which a mode of operation for the calculation of the data threshold level 146 is determined. The method 400 is performed by the averaging logic 218. Specifically, dependent upon the seven-bit output sequence of the noise

follower 194 (that is representative of the noise threshold level 146), the averaging logic 218 may operate in a first mode wherein the data threshold level 146 is calculated to be the average of the peak and noise threshold levels 148 and 144, or in a second mode wherein the data threshold level 146 is calculated to be half of the peak threshold level 148. Referring now to the flowchart in **Figure 26**, the method 400 commences at step 404, whereafter a determination is made at decision box 406 by the averaging logic 218 as to whether the noise threshold level 146, as represented by the seven-bit output sequence of the noise follower 194, is less than, or equal to, a predetermined minimum floor value. If the noise threshold level 146 is less than, or equal to, the predetermined minimum floor value (e.g., less than 50 mV), the averaging logic 218 then enters the second mode of operation, wherein the data threshold level 146 is calculated to simply be half of the peak threshold level 148. Alternatively, the averaging logic 218 enters the first mode of operation, wherein the data threshold level 146 is calculated to the average of the peak threshold level 148 and the noise threshold level 144.

It is advantageous to calculate the data threshold level 146 as half of the peak threshold level 148 if the noise threshold level 146 is less than, or equal to, the predetermined minimum floor value. Specifically, for very small data signals, the noise floor may introduce an artificial offset if the data threshold level 146 is calculated to be the average of the peak and noise threshold levels 144 and 148. In this case, the data

005710-091598

threshold level 146 may be located at a level at which the slope of the envelope signal 142 is not very steep, and accordingly time-jitter may become a consideration. Accordingly, in the event that there is a very little noise on the carrier medium, it is beneficial to locate the data

5 threshold level 146 at 50 percent of the peak threshold level 148, and not to factor in the noise signal (using the noise threshold level 142).

However, for larger signals, the average of the noise and peak threshold levels 144 and 148 may provide an improved result.

Thus, a method and apparatus for automatically determining a  
10 peak voltage level for a data signal propagated on a carrier medium have been described. Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention.

15 Accordingly, the specification and drawings are to be regarded in an illustrative, of than a restrictive, sense.